Order this document
by BR1107/D

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

LONWORKS™
Applications
Primer

MOTOROLA

Contents

Introducing LON Technology

Local operating network (LON) technology offers a powerful means for implementing distributed systems that perform sensing, monitoring, control, and other applications. LONWORKS™, a collection of tools and components developed by Echelon, can be used to build specific LON applications. LONWORKS enables the design of low-cost, communicating, and cooperating products that can be linked in a wide variety of ways to automate buildings, factories, vehicles, homes, and machines.

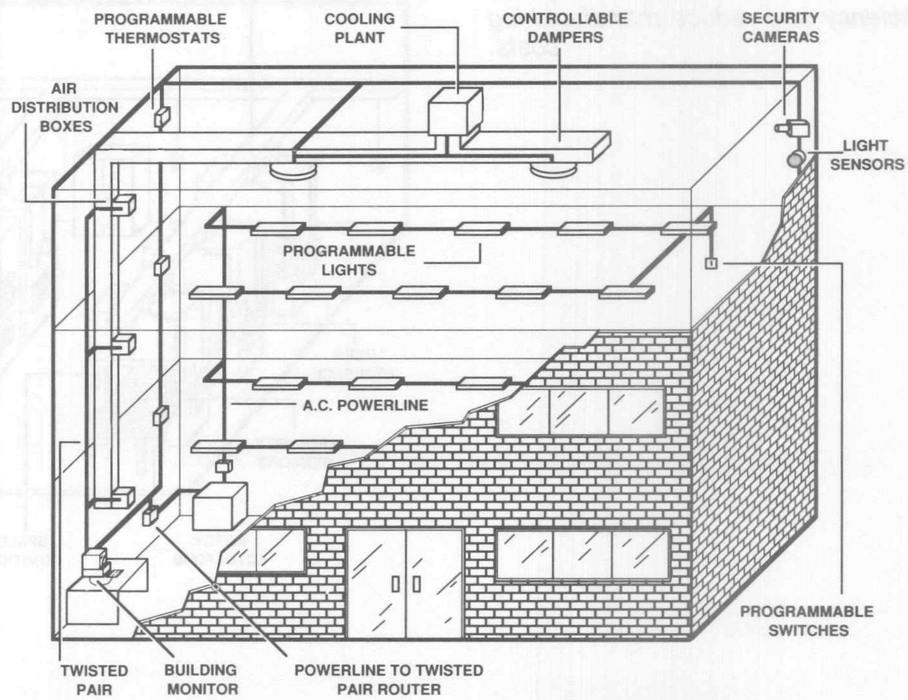
The Difference Between LONs and LANs

LONs constitute a class of technology that allows intelligent devices, such as actuators and sensors, to communicate with one another through an assortment of communications media using a standard protocol. In this way, a LON is similar to a local area network (LAN) that allows office computers to communicate with one another. LON technology supports distributed, peer-to-peer communications: individual network devices can communicate directly with one another, and a central control system is not required.

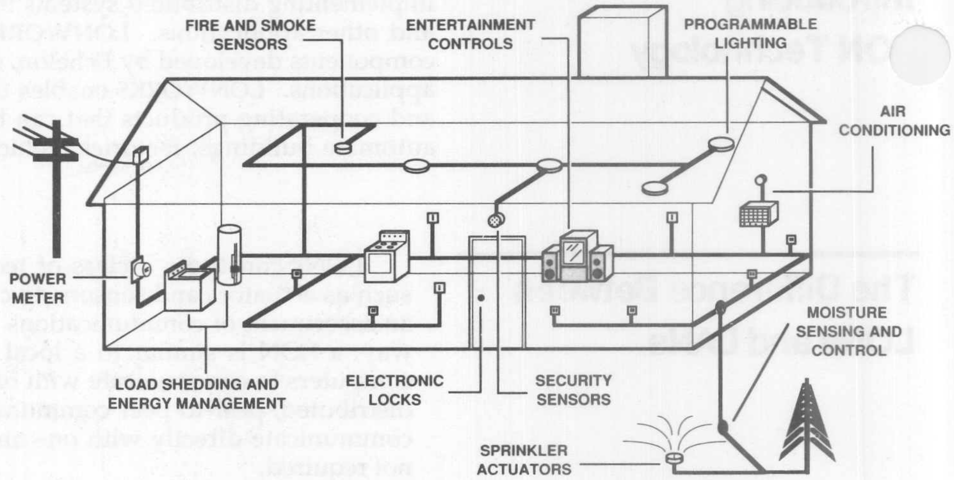
What distinguishes LONs from LANs is their purpose. LANs are designed to move data (such as documents, images, and databases) among computers, shared disks, and printers. A LAN's performance is viewed in terms of its throughput, usually measured in megabits transmitted per second. A LON is designed to move sense and control messages which are typically very short and which contain commands and status information that trigger actions. LON performance is viewed in terms of transactions completed per second and response time.

Some potential applications for LON technology are shown in the following illustrations.

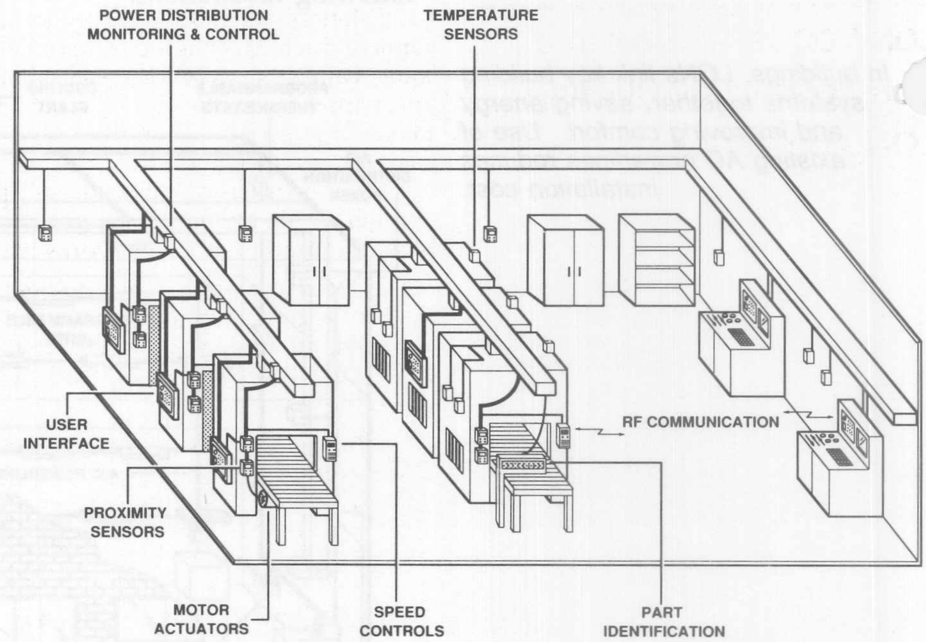
In buildings, LONs link key building systems together, saving energy and improving comfort. Use of existing AC powerlines reduces installation cost.



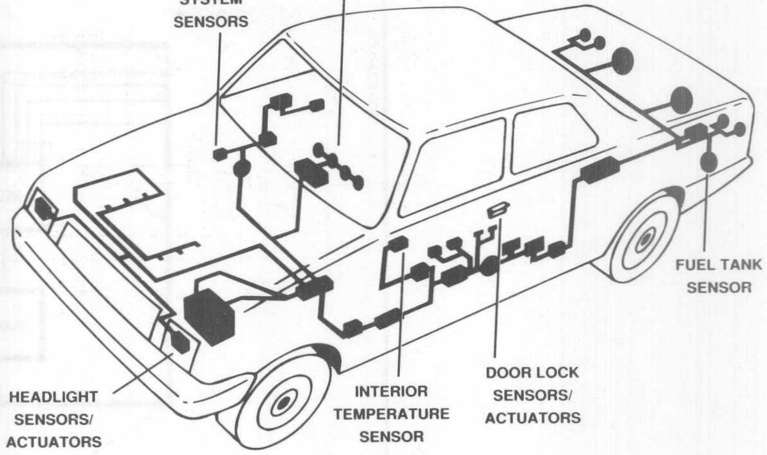
LONs make homes smarter, safer, and more convenient without the need for rewiring.



Factories can be programmed and reprogrammed quickly. Systems share information to improve efficiency and reduce manufacturing costs.



In vehicles, LONs eliminate thousands of feet of wiring, reducing weight, improving fuel efficiency, increasing reliability, and making vehicles easier to manufacture.



LONWORKS — Tools and Components for Building LON Applications.

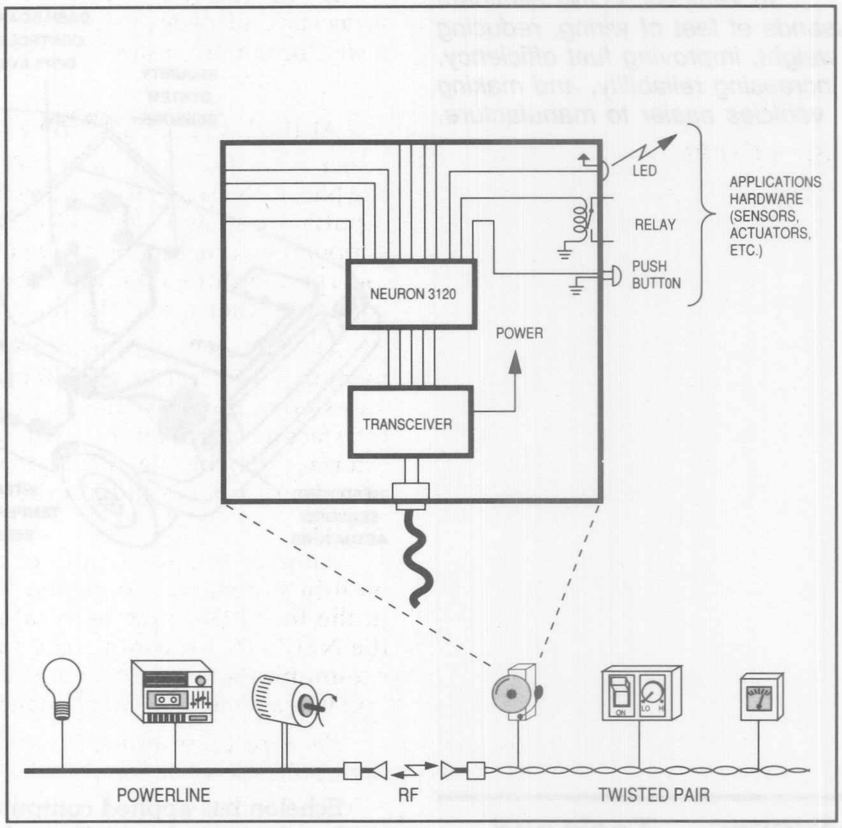
Echelon has applied computer, communications, control, and semiconductor technologies to develop a collection of components and tools, called LONWORKS, that enables manufacturers to develop, deploy, and support systems and products based on LON technology. LONWORKS makes it possible to build many kinds of products that are cost-effective, easy to install, and flexible. Moreover, different manufacturers may elect to design LONWORKS-based products that are compatible with one another. This opens up the prospect of new markets and applications based on multi-vendor networks made up of smart, interoperable products.

LONs are made up of intelligent, communicating devices, called *nodes*, that are logically combined. A node may be a basic wiring device such as a switch, an analog or digital sensor, an actuator, or even a user interface device like a display or keypad. LONWORKS nodes typically consist of the following:

- one or more sensors or actuators and associated application hardware.
- a NEURON CHIP™.
- a transceiver that provides the link to the communications medium.

The following diagram depicts a typical LONWORKS node:

LONWORKS supports a variety of devices on multiple communications media.



The Elements of LONWORKS

LONWORKS includes all the elements required to design, build, and maintain LONs: NEURON CHIPs and LONWORKS transceivers that form the physical parts of the node; the LONTALK™ protocol that provides reliable communications and the basis for interoperability among products; and the LONBUILDER™ Developer's Workbench that enables rapid development of products and networks.

The LONTALK Protocol

Echelon developed LONTALK as the framework for all the communications that occur on the LON. The LONTALK protocol is versatile and supports a wide range of applications, from consumer electronics, to factory automation, to vehicle controls, to commercial building controls, and to home automation. LONTALK is a complete 7-layer protocol as defined by the International Standards Organization (ISO) Open Systems Interconnection (OSI) reference model.

The LONTALK protocol is designed to provide reliable delivery of short control messages across a variety of media. With the appropriate transceiver, nodes can communicate with one another through virtually any medium, including twisted pair, powerline, and radio frequency.

The protocol also defines layer 7 — called the *application layer* — that enables manufacturers to build interoperable products.

LONs that follow the LONTALK protocol are called *LONWORKS networks*, and nodes using the LONTALK protocol are called *LONWORKS nodes*, or simply *nodes*.

NEURON CHIPS

At the heart of each node is a member of the NEURON CHIP family, either a NEURON® 3120 or NEURON® 3150. Both the NEURON 3150 and the NEURON 3120 provide a single-chip solution for communications, input/output, and control for many applications; the NEURON 3150 also supports external memory for more complex applications. Both incorporate three identical 8-bit CPUs; two are dedicated to communications, and the third is dedicated to application code.

The NEURON 3120 incorporates on-board RAM, ROM, and EEPROM memory. The NEURON 3150 has RAM and EEPROM on-chip and provides an external memory interface. Both have 11 bi-directional I/O pins for the interface to applications hardware, a 5-pin LONWORKS transceiver interface, flexible timer/counter hardware, and many other features, including a unique 48-bit serial number permanently programmed into every device.

Another unique feature of NEURON CHIPS is that they come with built-in firmware. Contained within the NEURON 3120's 10K of ROM (or in the first 10K of the external memory addressed by the NEURON 3150) is the NEURON firmware. The firmware implements the full LONTALK communications protocol and handles all the details of task scheduling, I/O management, network management, communications, and housekeeping.

Because these functions are managed automatically by the firmware, the applications developer is free to concentrate on *what each node will accomplish and its interaction with other nodes.* This allows developers to spend their time on creative approaches to applications, rather than focusing on the low-level details of communications and control.

LONWORKS Transceivers

A key feature of LONWORKS networks is their ability to communicate across different types of transmission media in a single network. The following applications put this into perspective:

In a factory warehouse, a forklift approaches a conveyor belt to pick up a pallet destined for outside shipment. As the forklift approaches, the conveyor belt is activated to position the pallet properly for loading, then turns off when the pallet is loaded. At the same time, a record is entered in the inventory control system to track the movement of the pallet from the warehouse to the shipping department.

In this application, the forklift contains a node that uses radio frequency for communication with the conveyor belt node. In turn, the conveyor belt node could use twisted pair media to communicate with power actuators and with interfaces to the inventory control system.

Another application might be as follows:

An intruder enters a secured area of a building after hours. A motion detector connected to other motion detectors with twisted pair wiring sends a message to the powerline-based lighting system to illuminate the intruder's area and sends a message by radio frequency to sound the alarm at the security entrance to the building. The message sent by the motion detector is also received, over the powerlines, at the security gate, which closes to seal off the area so the intruder cannot exit.

The ability for nodes to communicate across a variety of media is supported by a family of low-cost, high-throughput transceiver products for LONWORKS networks. The first products, developed by Echelon, are a set of evaluation modules that allow NEURON CHIPs to communicate over powerlines, radio frequency, and twisted pair wiring. In addition, the NEURON CHIP's flexible communications port enables developers to implement transceivers for other media (e.g. coax, fiber optic, etc.) to meet special needs.

LONBUILDER Developer's Workbench

The LONBUILDER Developer's Workbench is a powerful set of hardware and software tools for creating LONs. It operates in conjunction with an IBM® PC/AT® or compatible. LONBUILDER allows the developer to use a PC to begin development with a pair of nodes and grow to networks with as many as 24 emulator-based nodes and 256 remote nodes.

The LONBUILDER Developer's Workbench includes a complete collection of development-oriented hardware components, all of which can be housed within a multi-slot LONBUILDER Development Station. The Development Station includes a Control Processor that supports a high speed link to the host PC (and additional Development Stations); it also provides a network connection for the Protocol Analyzer and for the Network Manager. In addition, the six unassigned slots in the Development Stations can house any mix of the following LONBUILDER Processor Boards:

- **NEURON Emulator** for source-level software debugging and hardware prototyping.
- **LONBUILDER SBC** (single board computer) for use at the development PC or remotely with a transceiver and user-supplied power supply.
- **LONBUILDER Router** to provide communication between two different channels or media (e.g., twisted pair wiring and radio frequency).

LONBUILDER Transceiver Evaluation Boards provide the physical interface between an Emulator, SBC, Router, or Control Processor and a network channel.

The LONBUILDER Developer's Workbench also provides developers with a powerful set of integrated software tools:

- **LONBUILDER Integrated Development Environment** includes an object database, a project manager, and an integrated editor.
- **NEURON C™ Developer's Kit** contains the NEURON C compiler and the NEURON C source-level debugger.
- **LONBUILDER Network Manager** allows programmers to load, configure, and control LONWORKS networks.
- **LONBUILDER Protocol Analyzer** allows programmers to monitor and diagnose network traffic.

NEURON C — a high-level programming language based on the ANSI C language — that enables developers to rapidly produce applications that involve scheduling, node-to-node communications, and hardware interfaces. The NEURON C applications, in turn, rely on the firmware to manage and implement the details of how the developer's scheduling, communications, and hardware interface requirements actually translate to chip-level activity. In this way, the firmware provides a network operating system environment for executing the developer's application program and frees the developer from dealing with low-level details of communications and input-output processing. Combining NEURON C and the NEURON firmware together facilitates compact programs that are easy to develop and maintain.

LONWORKS networks provide the capability to connect a variety of nodes performing distributed control, sensing, identification, or other applications. The nodes use one or more communications media and communicate with one another using a common protocol, LONTALK. Devices on the LONWORKS network are programmed in NEURON C so that they will send, receive, and act on messages or changes in conditions. Developers concentrate only on *what nodes on the network must do,* and instructions built into NEURON CHIPs at each node take care of *how* nodes will carry out these instructions.

A LONWORKS-Based Lighting Control System

A simple lighting control system for one floor of a building illustrates how a LONWORKS-based system can be used. Unlike centralized systems, a system based on LONWORKS can be installed cost-effectively one floor at a time. This example is germane because it involves both design and installation issues. At the end of this brochure, some additional types of systems supported by LONWORKS that have different installation requirements are discussed.

The building floor in the example has a number of offices, some with windows and some without. Part of the control system's function is to respond to occupancy — the lights are turned on when someone enters an office and turned off when no one is there. For offices with windows, the system must additionally turn the lights off (regardless of occupancy) whenever there is enough natural light from the windows.

The overall design approach with LONWORKS is to define the nodes required in the system, define the connections between nodes, and write application code for each node to tell it what to do. With proper design, the nodes become generic building blocks that can be applied in various ways to accomplish various tasks (e.g., to control the lighting on many different floors of many different buildings using a variety of communications media). The tasks the nodes perform in any given situation are determined by how they've been connected and configured.

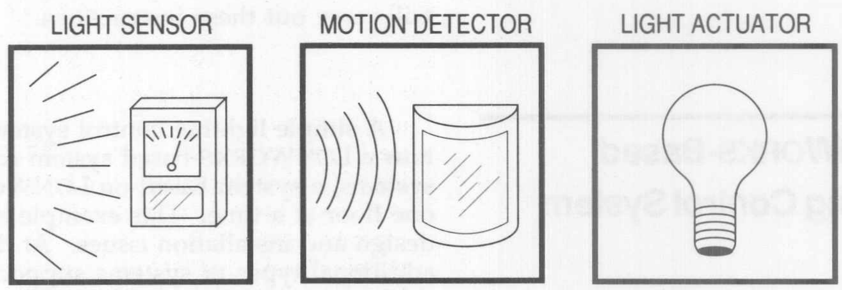
Because hardware design, software design, and network design are all independent in a LONWORKs -based system, a node's functions can be programmed without concern about the specifics of the networks in which they will be used.

Defining the Nodes in the System

In this example, there are three types of nodes: light sensor nodes, one for each side of the building; motion detector nodes, one for each office; and light actuator nodes, one for each lighting fixture. To minimize the need for new wiring, network communication will be via the existing AC powerlines that supply power to the lights. Of course, the design would proceed in the same manner if, for example, twisted pair communications were chosen.

Physically, each node will consist of a NEURON CHIP, a powerline transceiver, either a sensor (light sensor or motion detector) or a light actuator, and whatever interface circuitry might be required between the NEURON and the sensor or actuator.

Logically, the nodes can be thought of as black boxes.



Defining Connections Between Nodes

After the various types of nodes have been defined, the next step is to define the connections between them. Nodes in a LONWORKS network are connected via objects called *network variables* (NVs).

Whenever a network variable is assigned a value within a node's application program, this value is automatically sent to all other nodes on the network that have been configured to receive this data. Each type of NV can be defined as an input or output object. A node with an output NV of any given type has the *potential* to communicate with all other nodes on the network that have input NVs of the same type. For example, if a node has a Kilowatt-Hours *output* NV, it has the potential to communicate with all the nodes on the network that have a Kilowatt-Hours *input* NV.

Network variables can represent a wide range of quantities: integers, Boolean values, character strings, etc. Developers have complete flexibility in defining the NVs used in their programs and can even define network variables that are structures containing multiple data types. In addition, Echelon also offers a predefined set of Standard Network Variable Types (SNVTs). SNVTs have additional properties that make installation easier and also promote interoperability among different products. LONTALK will support as many as 255 SNVTs, although a wide range of applications can be served using the 50 or so SNVTs defined thus far. The definition of a SNVT includes units, a range, and an increment. Here are some examples:

Name	Quantity	Units	Range	Increment	ID #
SNV_Deg_F	Temperature	Celsius	-3200 to +3200	0.1 degree	1
SNV_Time_sec	Elapsed Time	Seconds	0 to 65,000	1 second	12
SNV_events	Event Count	Counts	0 to 65,000	1 count	4
SNV_KWH	Energy	Kilowatt-Hours	0 to 650	0.01 KWH	17
SNV_Lph	Flow	Liters/hour	0 to 65,000	0.1 lph	21
SNV_inHg	Pressure	Inches-Hg	-320 to 320	0.01 in HG	37
SNV_switch_state	Switch State	Boolean	Open(F) Closed (T)	N/A	3
SNV_phone_state	Phone Status	Enumerated	On-hook, off-hook, ringing, busy, ...		45
SNV_string	Character string	ASCII characters	0-31 characters		40

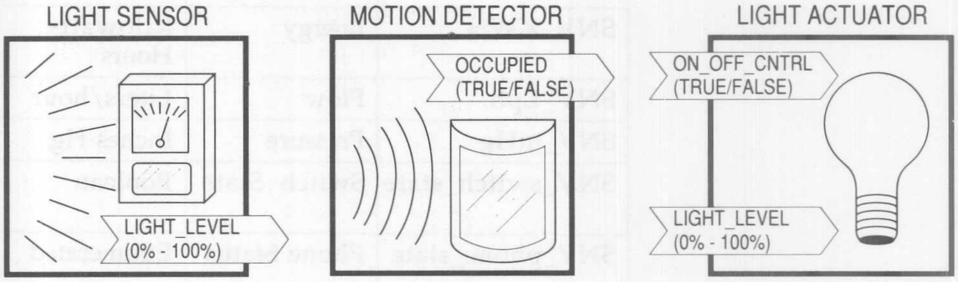
A developer is not required to use SNVTs. Network variables of any arbitrary type may be defined. However, if SNVTs are used, the developer of a LONWORKS node has the option of enabling nodes to identify and document their network variable inputs and outputs *over the network*. This is accomplished by storing within the node two key pieces of information about each of the node's network variables — the SNVT ID number and a text string. Using standard network management commands, a LONWORKS node can extract the SNVT information from any other node. Thus, a query of the light actuator node would return the following information over the network:

Node Type: Light

Network Variable Number	1	2
Input or Output	Input	Input
ID #	3	8
Type	Switch State	% of Scale
Units	Boolean	0-100%
Increment	–	0.4%
Description	"Light on/off control"	"Ambient Light Level"

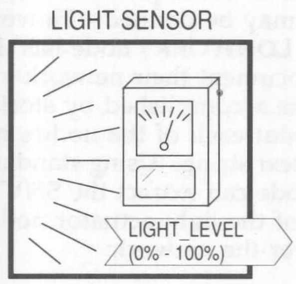
The ability to extract this information from the nodes themselves can greatly simplify installation and maintenance of LONWORKS nodes and networks.

By convention, input NVs are indicated graphically as arrows entering a node, output NVs by arrows exiting a node. The following example uses a %-OF-FULL-SCALE output SNVT for the light sensor, a TRUE/FALSE output SNVT for the motion detector, and corresponding input SNVTs for the light actuator. With the appropriate SNVTs, the nodes in the example look like this:



Programming the Nodes

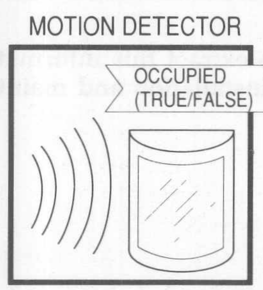
After the nodes and connections have been defined, the next step is to program them. For the light sensor node, the instructions are something like this:



When the light level changes by 2% or more than 20 minutes has elapsed since the last update, send out a new light level update.

The light sensor node produces an output representing the light intensity. This output can be represented in several ways, for example, as lumens, or some other standard measure of brightness. Another interesting way to represent the light level is as a percentage of full outdoor daylight. In this case, the light sensor's output would be a value between 0% and 100%.

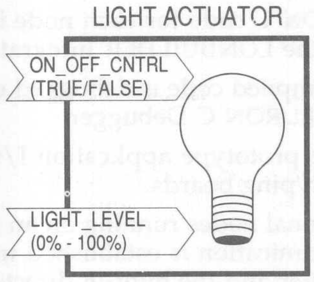
For the occupancy (motion detector) node, the instructions generate a Boolean output:



When someone is detected entering the room, send out the update, OCCUPIED = TRUE; when it is detected that the room is unoccupied, send out the update, OCCUPIED = FALSE.

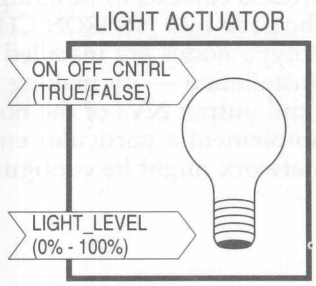
The instructions for the light actuator node are somewhat more complex, requiring decisions based on how the system is designed to operate. For example, if the lighting system has physical switches that someone can turn on and off, would human or occupancy sensor input prevail? Would the system allow someone with a window office to turn on the lights on a sunny day or prohibit this possibility? In distributed systems, it is normally good design practice to resolve conflicts such as these at the point of actuation.

For the purpose of this example, the light actuators are automatic. The instructions for the light actuator in a windowless room are the following:



If OCCUPIED = TRUE, turn the light on; if OCCUPIED = FALSE, turn the light off.

Obviously, this is not a complete solution, since it is also necessary to take into account the offices with windows. Does this mean a new type of node is required — one containing a NEURON CHIP with a different program? Not with LONWORKS. Creating a new type of node can be avoided by giving the light actuator nodes all the instructions they need to handle both situations (offices with or without windows) and later establishing the network connections so that each light actuator node receives only those inputs it needs to act upon. In the example, the instructions for the light actuator would be the following:



If OCCUPIED = TRUE and the ambient light is below the lighting level setpoint, turn the light on; else turn the light off.

These instructions cover all the contingencies of the example. In a more complex system with more contingencies, it might be necessary to have more than one type of actuator node. In any case, the decision on how to partition a LONWORKS network, like any partitioning decision, is based on the specific requirements of the installation and cost. Even simple nodes can perform complex functions when properly connected. Often, additional functions can be added to a node without additional cost.

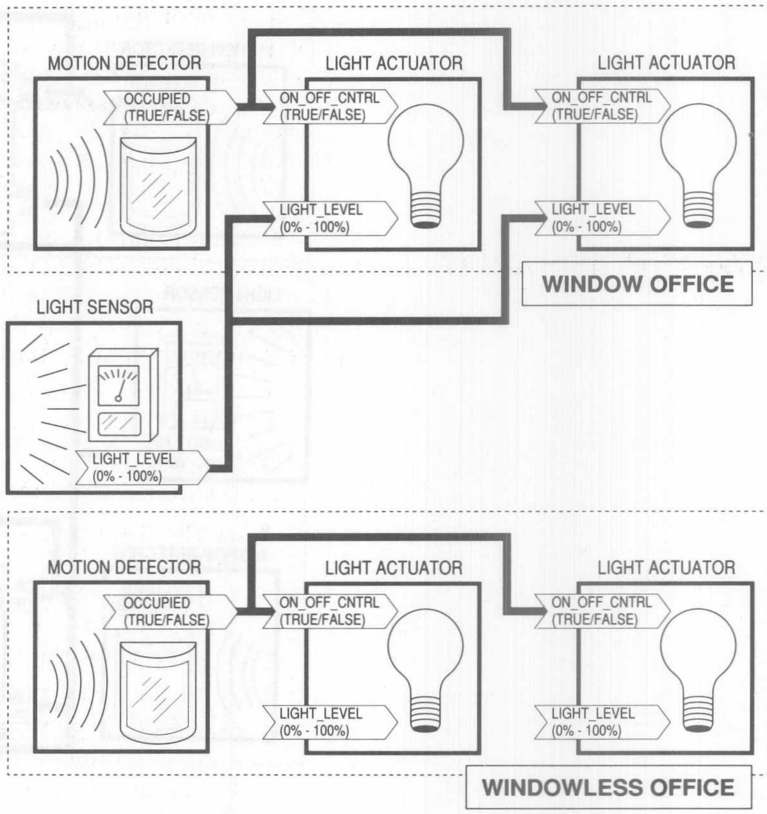
Additionally, the *lighting level setpoint* referred to in the previous instructions for the light actuator node could be a fixed number determined at the time the node was designed, a value set by a technician during installation, or even a level adjusted by an end-user via a knob or slider on either the light actuator node itself or another remote node. For the moment, it is a fixed number, 60% of full scale, established while writing the node's application program.

Implementing the Nodes

With the programs of one or more nodes defined, the next step is to implement them. The LONBUILDER Developer's Workbench provides all of the hardware and software tools needed to compile the program for each node, test it, and then configure and test networks of nodes. A typical development scenario for a powerline application is as follows:

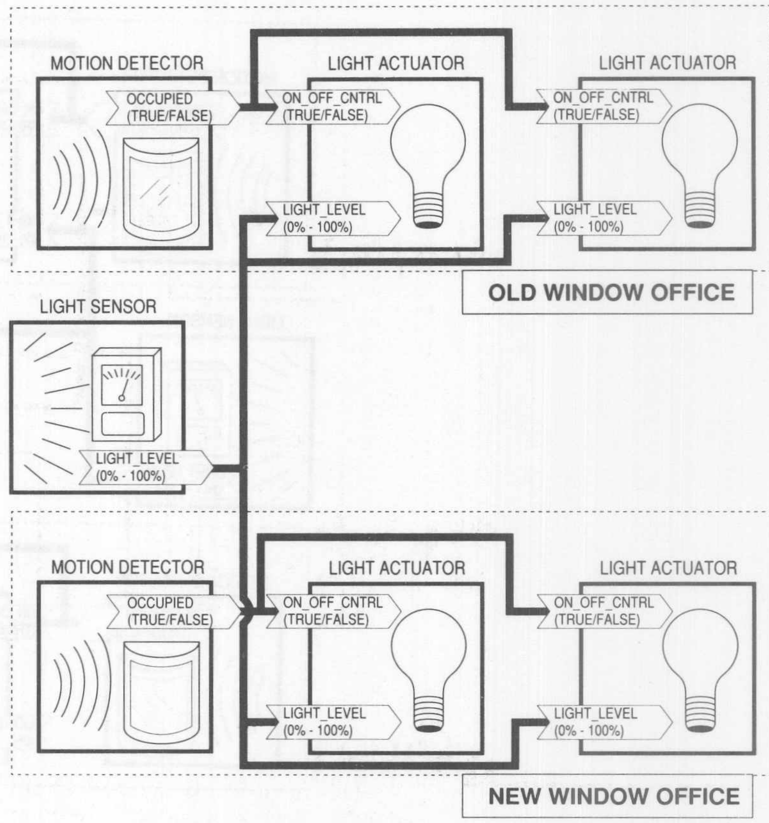
- NEURON C code for each node is written, edited, and compiled with the LONBUILDER Integrated Development Environment.
- The compiled code is debugged on the LONBUILDER Emulator using the NEURON C Debugger.
- Simple prototype application I/O hardware is implemented on I/O prototyping boards.
- Additional nodes running on an Emulator are added and communication is established using the LONBUILDER Network Manager and the built-in direct-connect network.
- Network traffic is viewed on the LONBUILDER Protocol Analyzer.
- The code for each node is loaded on a LONBUILDER single board computer (SBC). Each SBC is connected to a Powerline Transceiver Evaluation Unit and is plugged into the AC powerline. The Protocol Analyzer and Network Manager nodes on the LONBUILDER Control Processor card are also connected to the powerline via a powerline transceiver evaluation unit.
- Various configurations of nodes are set up and tested.

Once the applications code for each type of node has been written and debugged using LONBUILDER emulators and SBC's, the next step in the development process consists of building prototype hardware and downloading the code into NEURON CHIPs on each prototype node. At this point, prototype nodes are installed and configured as they would be in a customer installation — by making the appropriate connections between input and output NVs of the nodes. This is how generic nodes are configured to implement a particular end-use application. In this example, the prototype network might be configured as follows:

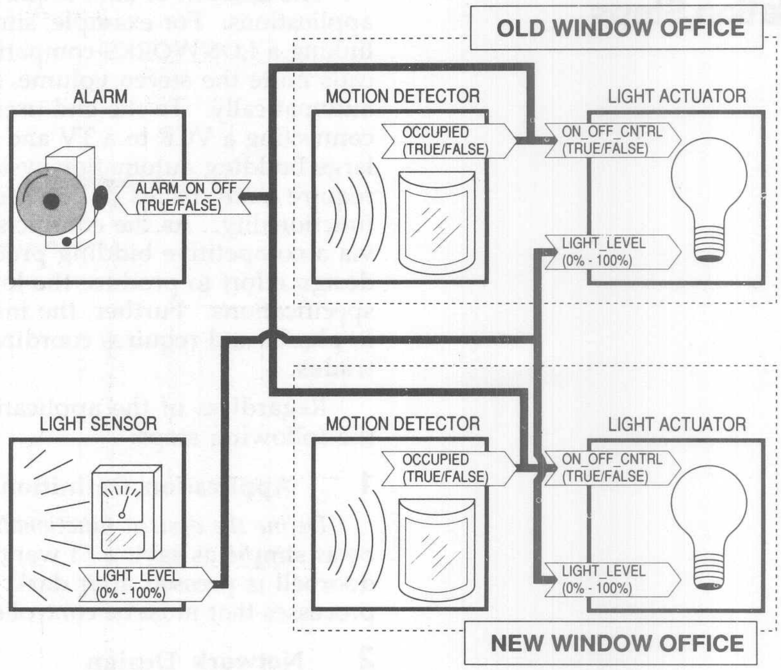


This figure shows that for window offices, the *light level* output NV of the ambient light detector and the *occupancy* output NV of the motion detector are "*connected*" to the light actuators' corresponding *on/off control* and *light level* input NVs. The connection, of course, is logical rather than physical: the same physical communications path can support many logical network variable connections. In windowless offices, only a motion detector's *occupancy* output NV is connected to the light actuators' corresponding *on/off control* input NV. If a node has an input NV that never gets connected to an output NV, then no updates are received and the code associated with updates to that input NV does not execute.

Because network parameters — established in the process of connecting network variables — are independent of node application code, reconfiguring the network is as simple as making new NV connections among the affected nodes. For example, if new tenants moved into a building and added windows to formerly windowless offices, then the lights in these offices could be made to respond to ambient light by establishing a connection between the affected light actuator nodes and the light sensor. In the five-node prototype, the new configuration would look like this:



Once installed, the nodes could become components in any other LONWORKS-compatible network. For example, the motion detector nodes could be connected to an alarm system programmed to sound an alarm (or transmit a message to a law enforcement agency) whenever motion was detected in certain rooms during certain hours of the day. As long as the alarm node had a SNV_SWITCH_STATE input NV, it could be connected to the previously installed motion detector nodes by a simple operation. This is how the prototype network would look with the alarm added.



LON Installation

The network described in the lighting example could be installed in several different ways. For example, if the lighting system were small, consisting of only the two offices and fewer than ten nodes, it might be installed by building maintenance personnel. If the system were part of a complete building-wide lighting system retrofit involving hundreds of nodes, it would probably be installed by an electrical contractor. If the same nodes were used to automate lighting in a home, the homeowner might try a do-it-yourself approach.

The applications described in the previous examples share a number of key similarities, and exhibit some important differences. What is common to all scenarios are the basic steps required to implement the system once the desired functionality can be described. What is different is *where* each step is performed and the *level of skill* possessed by the person who does the work.

Installation Steps

The amount of skill required to install a LON can vary widely among applications. For example, simple home automation functions, such as linking a LONWORKS-compatible stereo to a telephone so that incoming calls mute the stereo volume, require that the installation steps occur automatically. To the end-user, the problem should be less complex than connecting a VCR to a TV and pushing a few buttons. On the other hand, large building automation systems with thousands of nodes will usually require many pages of detailed specifications to describe the desired functionality. As the contracts for these systems are normally awarded via a competitive bidding process, bidders will often invest substantial design effort to produce the lowest cost system that meets the system specifications. Further, the installation of these systems usually occurs in phases and requires coordination of multiple contractors from different trades.

Regardless of the application, all LONWORKS installations require the following steps:

1 Application Definition

Define the desired functionality as experienced by the end-user. This could be as simple as saying, "I want my porch light to turn on whenever my doorbell is pressed after dark," or it could be a detailed description of processes that must be controlled on a production line.

2 Network Design

Select nodes and define where they will be physically placed and how they will be logically connected. This involves defining the connections between the nodes' network variable inputs and outputs as well as selecting the media. In the home automation case, the porch light fixture would communicate over the powerline, and the doorbell selected could be an RF device to eliminate the need for new wiring. An RF-to-powerline router would be plugged into a convenient outlet to transfer messages between the powerline and RF media. In a factory automation system, a high-speed backbone could link a collection of twisted-pair channels.

3 Physical Placement and Attachment

Locate nodes in their proper places and make any necessary connections to application hardware and to network media. This step proceeds in the same way with LONWORKS nodes as with *dumb* products: devices are mounted and plugged into (or wired to) their corresponding medium.

4 Node Customization

Load nodes with information that establishes the desired logical connections to other nodes. This is the process by which a generic node takes on a unique personality within the context of a specific network. Standard network management commands are used to load a node, via its network connection, with addressing information that defines the node's place in the network and its connections with other nodes. This information is called the node's *network image*.

Installation Scenarios

The lighting system, home automation, and factory automation examples shown on the first few pages of this document are all instances of networks that are composed of a collection of generic nodes linked in a unique configuration at a customer's site. There are other types of networks, such as multiplexed wiring systems that are wholly embedded within a vehicle or a piece of machinery in which the entire network is installed and tested at the factory. One helpful way of categorizing different installation scenarios is to focus on where, and by whom, the key steps of Network Design and Node Customization are accomplished. The following table shows one such classification:

NETWORK DESIGN PERFORMED ...	NODE CUSTOMIZATION PERFORMED ...	INSTALLATION CLASS
Intuitively	By the homeowner	PLUG & PLAY Supports mass market products that can be installed and configured by anyone.
In the factory	In the factory	EMBEDDED SYSTEMS LONs contained wholly within a product, such as a milling machine, airplane, office copier, etc.
By a captive or independent systems integrator, using network design guidelines or tools	By a captive or independent systems integrator or their subcontractors at a field office or on-site	ENGINEERED SYSTEMS Building automation systems, factory automation systems, and others that are typically addressed via a competitive bidding process
Ad hoc, either by a systems integrator (light commercial) or end-user (residential)	By on-site worker or end-user	DO-IT-YOURSELF Networks assembled from generic products sold through broad-line or specialty distributors, or mass market retailers

How LONWORKS Supports Different Installation Scenarios

Installation considerations have been central to the design of LONWORKS tools. These include the following:

- Each NEURON CHIP contains a unique 48-bit ID. This ensures that every node in a LONWORKS network has a unique address. Any NEURON-based node will respond to a packet addressed to its NEURON CHIP ID. This type of addressing is typically used only during installation. After installation is complete, nodes are typically addressed by logical address.
- Every NEURON CHIP also supports a special I/O pin called the Service Pin. When enabled, this pin can directly drive an LED, which indicates the node's status from the network point of view. For example, the service LED on an unconfigured node (i.e. one that has not received its network image) may blink in a particular pattern. In a *healthy* node, the Service LED is extinguished; in a *failed* node, the Service LED will remain lit. In addition, grounding the Service Pin causes a node to transmit a specially formatted message that contains the NEURON's 48-bit ID. This enables a node to identify itself over the network after the node has been physically installed.
- Nodes designed using Standard Network Variable Types (SNVTs) can offer self-identification and self-documentation capabilities. This allows nodes to identify their interface and functionality over the network.
- The LONTALK protocol includes a rich set of network management commands that are part of the NEURON firmware and are recognized by all LONWORKS nodes. These commands are used to load nodes with their network images, take nodes online and offline, reset and restart their application programs, report network performance statistics, perform diagnostics, and execute other functions that assist in network maintenance and trouble-shooting. Other commands are used to extract the self-identification and self-documentation data stored within nodes that use SNVTs.
- Through a set of LONWORKS application programming interfaces (APIs), developers can implement custom programs on a range of platforms that support their unique installation and network management requirements. The APIs support standard access to the LONWORKS network management functions. They also export some of the capabilities included in LONBUILDER, such as the network database manager and the *binder*, which generates node customization data automatically, given a list of desired network variable connections.

SUMMARY

LONWORKS™ networks offer the capability for manufacturers, developers, technicians, and end-users to connect an unlimited variety of devices to one another in a wide range of applications. These devices can be interconnected using communications media appropriate to the application and installation environment; they use a common protocol, LONTALK, that lays the foundation for interoperable products from different manufacturers — adding value to each manufacturer's product line.

Because of the flexibility of the technology, a broad spectrum of skill levels can be supported at the point of installation, customization and repair of LONWORKS network products. The preparation of the products for commercial installation, maintenance, and repair involves design decisions at the point of manufacture that take into account the nature of the product.

LON technology promises greater flexibility and modularity of product lines than ever before possible. Echelon's LONWORKS, including the NEURON CHIP, the LONTALK Protocol, LONWORKS Transceivers, and the LONBUILDER™ Developer's Workbench, provides key elements that support building practical products and achieving profitability from this promise.